# Pentium® Pro Processor
# BIOS Update API Test (BAT)
# Utility ver 1.1

## Test Procedures Rev 1.2

Test Methodology....4
Software Version and Support....4
General Procedures....4
Description of BIOS Update API....4
Presence - Function 00h....6
Presence Test....6
Write BIOS Update Data - Function 01h....7
INVALID_HEADER (Invalid Header Version in Header Field)....7
INVALID_HEADER (Invalid Loader Version in Header Field)....8
INVALID_HEADER_CS (Invalid Checksum in Header Field)....9
INVALID_REVISION (Invalid Revision )....10
CPU_NOT_PRESENT....10
Update Loader Function....11
Write Valid File Test....11
Authenticity Test....13
Authentication Test....13
Write Corrupted Data Test....14
BIOS Update Control - Function 02h....14
Control Status Query....14
Update Disable Functionality....15
Read BIOS Update Data - Function 03h....16
Update Control Functionality....16
Read Update Block Test....16

## *Read This Section Before You Begin:*

## Test Methodology

The BIOS Update API Test utility (BAT) uses a set of special test BIOS Updates, based on the current Update Ids available, to test the support of BIOS Update API INT 15h interface in a target BIOS. It is not necessary that the latest Update ID availabile be used.
This test utility is intended to be used by BIOS developers to test their BIOS Update API support, and confirm it's compliance with the Pentium® Pro processor BIOS Update specification. It is assumed that the user is very familiar with the BIOS Update API specification (see *Description of BIOS Update API* below).
This utility was not intended to be used as a BIOS Update loader application in a production environment.

## Software Version and Support

At the time of this printing, the latest version of BAT.EXE (version 1.1) supports the following processors listed below , along with their associated Update ID used in the test:

| **Processor-Step** | **BIOS Update ID** |
|---|---|
| P6.0-C0 | c5h |
| P6.0-A0 | c5h |
| P6S-A1 | c5h |
| P6S-B1 | d1h |

## General Procedures

BAT.EXE is a menu driven test utility that requires no special training or information. Occasionally you may find that items #6 (Read Update Test ) and #9 (Processor Information) in the main menu useful to verify which processor(s) contain which Update Id and the contents of the Update Blocks in NVRAM. The utility will prompt the user for the name of a log file to store the results in before exiting the program.

## Description of BIOS Update API

With the introduction of the Pentium Pro Processor came the "BIOS Update". A BIOS Update is a data block (2048 bytes) that, when presented to the processor, correct specific processor errata. For all current steppings (revisions) of the processor, BIOS Updates are required to be presented to the processor by the BIOS at boot time. The BIOS Updates are encrypted such that only the specific update block intended for a specific stepping of the processor in the system will be accepted by the CPU. These updates are stored in the NVRAM that also holds the BIOS. The *Pentium Pro Processor BIOS Writers Guide* specifies an API that the BIOS must provide. This API functions as a standard INT15h interface between a calling utility and the BIOS, that can be used to load new BIOS Update blocks into the non-volatile memory. Without it, an end user would not be able to add any new BIOS Update blocks to their system without the OEM providing either a completely new BIOS or a custom loader.

The API is defined to use an INT 15 interface. Therefore, parameters for the calls are loaded into the processor registers prior to the software issuing an INT 15h. Listed below are functions that are available for the API. For complete details of the function call, please refer to the *Pentium Pro Processor BIOS Writer's Guide*.

| BIOS Update Function | Function Number | Description | Required/Optional |
|---|---|---|---|
| Presence test | 00h | Returns information about the supported functions. | Required |
| Write BIOS update data | 01h | Writes one of the BIOS Update data areas (slots). | Required |
| BIOS update control | 02h | Globally controls the loading of BIOS Updates. | Required |
| Read BIOS update data | 03h | Reads one of the BIOS Update data areas (slots). | Required |

# Presence - Function 00h

## Presence Test

***Test Description:***

This test will exercise the INT 15h function 00h call and check for the correct values returned into the appropriate registers. The call is made by initializing AX with the value D042h and BL with the value 00h, followed by INT 15h instruction as described by the BIOS Update API specification. For Multi-Processor (MP) systems, the number of update blocks available in NVRAM represented by the returned value Update Cnt in SI is compared to the actual number of unique steppings currently on the system under test. It is recommended that all sockets on the system are populated with as many unique processors steppings as the motherboard design was intended to support.

### ✓ *Before Running this Test Item:*

On MP systems, it is recommend that functional processor sockets are populated using all available steppings supported by the system and the latest version of this tool.

### *Test Procedure:*

1. From the ***main menu*** choose **"Presence Test"** by hitting the '1'on the keyboard.
2. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
3. Hit any key to exit to main menu.

### *Inspecting The Results:*

The screen output from this test will display the results of each item listed below. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 0 |
| AH | Return Code | 00h (SUCCESS) |
| AL | OEM Error | Not Tested |
| EBX | Signature Part 1 | "INTE" |
| ECX | Signature Part 2 | "LPEP" |
| EDX | Loader Version | 00000001h |
| SI | Update Cnt | Number of update blocks in NVRAM |

*Additional Check:*

BAT will also check the returned value of Update Cnt in SI and evaluate this value as follows:

***if*** Update Cnt >= Number of unique processor steppings on board ***then*** **PASS** (with no warnings)

***if*** Update Cnt < Number of unique processor steppings on board ***then*** **FAIL** (*"Number of storage available in NVRAM is less than the number of uniqueCPU steppings on board the system."*)

***if*** Number of Processors on Board > Update Cnt >= Number of unique processor steppings on board ***then*** **Warning** (*"Number of storage available in NVRAM is less than the number of processor sockets detected in the system."*)

# Write BIOS Update Data - Function 01h

## INVALID_HEADER (Invalid Header Version in Header Field)

This test presents the BIOS with a BIOS Update that has an invalid header version (00000000h) in the header block. In an MP system, the model/stepping version of the BIOS Update used in the test will be based on the Boot Strap Processor's (BSP) model/stepping detected on the system. The **INVALID_HEADER** and **CF flag = 1** is checked for. In addition, this test will verify that the BIOS Update ID did not get presented to the processor or stored into any of the available Update Block slots in NVRAM.

### ✓ *Before Running this Test Item:*

Make sure no BIOS Update ID's associated with the stepping/model of the processor on the system are stored in any of the Update Blocks in NVRAM. This can be manually verified by using **"Read Update Test"** from the main menu. On a MP system, make sure there are no BIOS Update ID's associated with the stepping/model of the BSP on the system. This may require reprogramming of the BIOS to empty out the Update Blocks or swapping in a processor with a stepping version that does not currently have a BIOS Update stored for it in any of the Update Blocks in NVRAM.

### *Test Procedure:*

1. From the ***main menu*** choose **"Write Update Error Parameter Test"** by hitting the '2' on the keyboard.
2. From the resulting ***sub-menu*** choose **"Invalid Header (Corrupt Header Version)"** by hitting the '1' on the keyboard.
3. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
4. Hit any key to exit back to the previous sub-menu.

### *Inspecting The Results:*

The screen output from this test will display the results of each item listed below. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 1 |
| AH | Return Code | 95h (INVALID_HEADER) |

*In addition, the following conditions are tested for:*

i.) The update ID was not loaded into processor.
ii.) The update ID presented to the BIOS did not get stored in any of the BIOS's Update Blocks in NVRAM.

## INVALID_HEADER (Invalid Loader Version in Header Field)

This test presents the BIOS with a BIOS Update that has an invalid loader version (00000000h) in the header block. In an MP system, the stepping/model version of the BIOS Update used in the test will be based on the BSP's stepping detected on the system. The **INVALID_HEADER** and **CF flag = 1** is checked for. In addition, this test will verify that the BIOS Update ID did not get presented to the processor or stored into any of the available Update Block slots in NVRAM.

### ✓ *Before Running this Test Item:*

Make sure no BIOS Update ID's associated with the stepping/model of the processor on the system are stored in any of the Update Blocks in NVRAM. This can be manually verified by using **"Read Update Test"** from the main menu. On an MP system, make sure there are no BIOS Update ID's associated with the stepping/model of the BSP on the system. This may require reprogramming of the BIOS to empty out the Update Blocks or swapping out a processor with the model/stepping version that does not currently exist in the Update Blocks in NVRAM.

### *Test Procedure:*

1. From the ***main menu*** choose **"Write Update Error Parameter Test"** by hitting the '2' on the keyboard.
2. From the resulting ***sub-menu*** choose **"Invalid Header (Corrupt Loader Version)"** by hitting the '2' on the keyboard.
3. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
4. Hit any key to exit back to the previous sub-menu.

### *Inspecting The Results:*

The screen output from this test will display the results of each item listed below.

The register values are inspected following the function 01h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 1 |
| AH | Return Code | 95h (INVALID_HEADER) |

*In addition, the following conditions are tested for:*

i.) The update ID was not loaded into the processor.
ii.) The update ID presented to the BIOS did not get stored in any of the BIOS's Update Blocks NVRAM.

## INVALID_HEADER_CS (Invalid Checksum in Header Field)

This test presents the BIOS with a BIOS Update that has an invalid checksum in the header block. In an MP system, the stepping model version of the BIOS Update used in the test will be based on the BSP's model/stepping detected on the system. The **INVALID_HEADER_CS** and **CF flag = 1** is checked for. In addition, this test will verify that the BIOS Update ID did not get presented to the processor or stored into any of the available Update Block slots in NVRAM.

### ✓ *Before Running this Test Item:*

Make sure no BIOS Update ID's associated with the stepping/model of the processor on the system are stored in any of the Update Blocks in NVRAM. This can be manually verified by using **"Read Update Test"** from the main menu. On an MP system, make sure there are no BIOS Update ID's associated with the stepping/model of the BSP on the system. This may require reprogramming of the BIOS to empty out the Update Blocks or swapping out a processor with the model/stepping version that does not currently exist in the Update Blocks in NVRAM.

### *Test Procedure:*

1. From the ***main menu*** choose **"Write Update Error Parameter Test"** by hitting the '2' on the keyboard.
2. From the resulting ***sub-menu*** choose **"Invalid Header Check Sum"** by hitting the '3' on the keyboard.
3. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
4. Hit any key to exit back to the previous sub-menu.

### *Inspecting The Results:*

The screen output from this test will display the results of each item listed below.

The register values are inspected following the function 01h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 1 |
| AH | Return Code | 96h (INVALID_HEADER_CS) |

*In addition, the following conditions are tested for:*

i.) The update ID was not loaded into the processor.
ii.) The update ID presented to the BIOS did not get stored in any of the BIOS's Update Blocks in NVRAM.

## INVALID_REVISION (Invalid Revision )

This test attempts to load a **valid** BIOS Update into the processor twice. The first time the update block is presented, an attempt is made to load the BIOS Update ID into the processor. The same Update ID is presented to the BIOS interface a second time to force an INVALID_REVISION condition to occur. In a MP system, the stepping /model version of the BIOS Update used in the test will be based on the BSP's model/stepping detected on the system. The **INVALID_REVISION** return code and **CF flag = 1** is expected and checked for.

### ✓ *Before Running this Test Item:*

Make sure no BIOS Update ID's associated with the stepping/model of the processor on the system are stored in any of the Update Blocks in NVRAM. This can be manually verified by using **"Read Update Test"** from the main menu. On an MP system, make sure there are no BIOS Update ID's associated with the stepping/model of the BSP on the system. This may require reprogramming of the BIOS to empty out the Update Blocks or swapping out a processor with the model/stepping version that does not currently exist in the Update Blocks in NVRAM.

### *Test Procedure:*

1. From the ***main menu*** choose **"Write Update Error Parameter Test"** by hitting the '2' on the keyboard.
2. From the resulting ***sub-menu*** choose **"Invalid Revision"** by hitting the '4' on the keyboard.
3. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
4. Hit any key to exit back to the previous sub-menu.

### *Inspecting The Results:*

The screen output from this test will display the results of each item listed below.

The register values are inspected following the function 01h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 1 |
| AH | Return Code | 98h (INVALID_REVISION) |

## CPU_NOT_PRESENT

This test attempts to load a BIOS Update associated with a model/stepping of processor that does not currently exist on the system. Because the BIOS is not able to complete the authenticity test in this given situation , the BIOS should not attempt to load the file into an available Update Block in NVRAM. The **CPU_NOT_PRESENT** return code and **CF flag = 1** is expected and checked for.

### ✓ *Before Running this Test Item:*

Decide on which stepping version of BIOS Update to be used in the test, and make sure that stepping version of the processor is currently ***not*** populated on the system board. Also, make sure that this BIOS Update ID does not currently exist in any of the Update Blocks in NVRAM. This can be manually

verified by using **"Read Update Test"** from the main menu. The contents in the Update Blocks can be determine by choosing option 6 (Read Update Test) from the main menu.

### *Test Procedure:*

1. From the ***main menu*** choose **"Update From a File"** by hitting the '8' on the keyboard.
2. From the resulting screen choose the stepping signature that associated with the processor that does not exist on board and is not currently stored in NVRAM.
3. Wait for response on screen and make note of the results. The BIOS should reject the test Update ID . The results on the screen will also be saved in a log file.
4. Hit any key to exit back to the previous sub-menu.

### *Inspecting The Results:*

The screen output from this test will display the results of each item listed below.

The register values are inspected following the function 01h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 1 |
| AH | Return Code | 94h (CPU_NOT_PRESENT) |

*In addition, the following conditions are tested for:*

i.) The update ID presented to the BIOS did not get stored in any of the BIOS's Update Blocks in NVRAM.

## Update Loader Function

### Write Valid File Test

**Description:**

This test involves performing an inventory of the model/steppings of all the processor(s) in the system, and querying the current BIOS Update ID loaded. This is accomplished by reading the BIOS signature using the RDMSR instruction for MSR B8H for each of the processor found on the system. The test application will next attempt to load a valid BIOS Update for the processor(s) that has been detected not to contain an update. Prior to writing any BIOS Update data to the processor, the update blocks in NVRAM are first checked for any BIOS Update ID's that should of been loaded during system boot. If a processor was found to contain no BIOS Update ID's after boot but a valid Update ID exists for it in NVRAM, a warning message will indicate an error. In a single or dual processor system where the BIOS supports 2 or less Update Blocks in NVRAM, the test should be repeated several times, swapping in a processor with a model/stepping value that has been purged out from NVRAM during the last update load attempt. The STORAGE_FULL (93H) should never occur.

## ✓ *Before Running this Test Item:*

This test requires a continuing swapping out of different processors to force loading of an associated test BIOS Update ID to exercise the BIOS's Update Block house cleaning algorithm. Since the current utility only provides one valid Update ID per processor signature, this test is most effective if the number of processor steppings used in the test exceeds the number of NVRAM Update Blocks supported by the processor. For example, if a BIOS supports 2 NVRAM update blocks in the system, a minimum of three different processor steppings should be used in this test (Note: At the time of the writing of this document, BAT.EXE supports four Pentium Pro processor signatures ( 612,616, 617, and 619). On an MP system, this test works best if all the processors on board are of the same stepping at any given time in the test.

## *Test Procedure:*

1. Populate the socket with a processor(s) that does not have an associated BIOS Update ID currently stored in any of the Update Blocks in NVRAM. To query the contents of the NVRAM, run BAT.EXE and choose item 6 from the main menu.
2. Run BAT.EXE and from the ***main menu*** choose **"Write Update Test"** by hitting the '3' on the keyboard.
3. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
4. Hit any key to exit back to the main menu.
5. Repeat steps 1 to 4 going through each stepping of processors available. The STORAGE_FULL (93H) in AH should never occur.

## *Inspecting The Results:*

The screen output from this test will display the results of each item listed below.

The register values are inspected following the function 01h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 0 |
| AH | Return Code | 00h (SUCCESS) |

# Authenticity Test

## Authentication Test

**Description:**
This test checks to insure that the BIOS correctly performs the authenticity check as prescribed by the *Pentium Pro Processor BIOS Writer's Guide*. This test is done by presenting the BIOS with a seemingly valid Update file reporting to have a numerically larger revision number in the header with a smaller revision number encrypted in the Update Data portion of the file. If performed correctly, the BIOS should obtain the signature from MSR 8Bh after the loading process and compare it to the Update Header to make sure they match. On a MP system, the file version associated with the BSP is used.

### ✓ *Before Running this Test Item:*

Make sure no BIOS Update ID's associated with the stepping/model of the processor on the system are stored in any of the Update Blocks in NVRAM. This can be manually verified by using **"Read Update Test"** from the main menu. On a MP system, make sure there are no BIOS Update ID's associated with the stepping/model of the BSP on the system. This may require reprogramming of the BIOS to empty out the Update Blocks or swapping out a processor with the model/stepping version that does not currently exist in the Update Blocks in NVRAM.

### *Test Procedure:*

1. From the ***main menu*** choose **"Write Authentication Test"** by hitting the '4' on the keyboard.
2. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
3. Hit any key to exit back to the main menu.

### *Inspecting The Results:*

The screen output from this test will display the results of each item listed below.

The register values are inspected following the function 01h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 1 |
| AH | Return Code | 97h (SECURITY_FAILURE) |

*In addition, the following conditions are tested for:*

i.) The update ID presented to the BIOS did not get stored in any of the BIOS's Update Blocks in NVRAM.

### Write Corrupted Data Test

**Description:**
This test checks to insure that the BIOS correctly returns SECURITY_FAILURE when a corrupted BIOS Update is rejected by the processor. If performing correctly, the BIOS will catch the failed return signature from the processor and return the correct failure code through the API. On a MP system, the BIOS Update associated with the BSP is used.

#### ✓ *Before Running this Test Item:*
Make sure there are no BIOS Update ID's associated with the stepping/model of the Pentium Pro Processor on the system stored in any of the Update Blocks in NVRAM. This can be manually verified by using **"Read Update Test"** from the main menu. On an MP system, make sure there are no BIOS Update ID's associated with the stepping/model of the BSP on the system. This may require reprogramming of the BIOS to empty out the Update Blocks or swapping out a processor with the model/stepping version that does not currently exist in the Update Blocks in NVRAM.

#### *Test Procedure:*
1. From the ***main menu*** choose **"Write Corrupted Data"** by hitting the '5' on the keyboard.
2. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
3. Hit any key to exit back to the main menu.

#### *Inspecting The Results:*

The register values are inspected following the function 01h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 1 |
| AH | Return Code | 97h (SECURITY_FAILURE) |

*In addition, the following conditions are tested for*

i.) The update ID presented to the BIOS did not get stored in any of the BIOS's Update Blocks in NVRAM.

## BIOS Update Control - Function 02h

### Control Status Query

**Description:**
This test checks to insure that the BIOS correctly returns the current status of the update loader (enabled or disabled). If disabling of the upload function is supported by the BIOS, the procedure will be included that requires the checking of the status after the load function is disabled.

### ✓ *Before Running this Test Item:*

No special requirements.

### *Test Procedure:*

1. From the ***main menu*** choose **"Control Update Test"** by hitting the '7' on the keyboard.
2. From the resulting sub-menu, choose "Query Status" by choosing '1' on the keyboard.
3. If supported by the system, reboot the system and disable write update in BIOS set-up.
4. Run BAT.EXE and repeat steps 1 through 3 above with the exception that a return value of "Disabled" in BL should be observed.
5. If the BIOS does successfully Disabled, try the Update From a File test to make sure that Update is functionally disabled.
6. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
7. Hit any key to exit back to the previous sub-menu.

### *Inspecting The Results:*

In summary, the values in the following registers are inspected following the function 01h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Entry | Description | |
|---|---|---|
| BH | Task | 2 = Query |
| Register on Return | Description | Expected Value |
| CF | Carry Flag | 0 |
| AH | Return Code | 00h (SUCCESS) |
| BL | Update Status | 1 = Enabled<br>0 = Disabled |

## Update Disable Functionality

**Description:**
This test checks to insure that the BIOS returns WRITE_FAILURE when a write is attempted while the Update Loader status = "disabled". This test is only applicable for a BIOS that allows the user to disable BIOS Updates through the BIOS setup.

***Control Status Enable Function***

**Description:**
This test checks to insure that the BIOS correctly enables the Update Loader after it has been previously disabled. This test is only applicable to systems which allows user to disable BIOS Updates through BIOS setup. If disabling of the Update Loader is not supported through BIOS setup, this function is still exercised to insure that the BIOS is not adversely affected by the function call.

### ✓ *Before Running this Test Item:*

No special requirements.

### *Test Procedure:*

1. If supported by the system, disable write update in BIOS set-up. Verify the status of the Update function by using the "Control Status Query" test.
2. From the ***main menu*** choose **"Control Update Test"** by hitting the '7' on the keyboard.
3. From the resulting sub-menu, choose "Enable Write Update" by choosing '2' on the keyboard.
4. Verify the control status returned the system to "Enabled" by following the procedure in the last section.
5. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
6. Hit any key to exit back to the previous sub-menu.

### *Inspecting The Results:*

In summary, the values in the following registers are inspected following the function 01h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Entry | Description | |
|---|---|---|
| BH | Task | 1 = Enable |
| Register on Return | Description | Expected Value |
| CF | Carry Flag | 0 |
| AH | Return Code | 00h (SUCCESS) |
| BL | Update Status | 1 = Enabled<br>0 = Disabled |

## Read BIOS Update Data - Function 03h

**Update Control Functionality**

### Read Update Block Test

**Description:**
This test checks to insure that the BIOS correctly performs the read function call as defined by the specification. The test application will query the number of update block in NVRAM that the BIOS supports by calling function - 00h Presence Test. The test application then reads each Update Block one at a time and reports back the Update Revision and model/stepping contained in each Update Block. The test determines if a specificUpdate Block is used or unused by reading the header version associated with the returned data. If a value of 0FFFFFFFFh is detected, the test will assume that particular Update Block is unused. If an invalid header version or any value other than 0FFFFFFFFh in the header version field is found, then an error is reported. A checksum on the returned Update Block data will also be performed to check the integrity of the data.

### ✓ *Before Running this Test Item:*

No special requirements.

### *Test Procedure:*

1. From the ***main menu*** choose **"Read Update Test"** by hitting the '6' on the keyboard.
2. Wait for response on screen and make note of the results. The results on the screen will also be saved in a log file.
3. Hit any key to exit back to the previous sub-menu.

### *Inspecting The Results:*

In summary, the following values are inspected following the function 03h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Return | Description | Expected Value |
|---|---|---|
| CF | Carry Flag | 0 |
| AH | Return Code | 00h (SUCCESS) |
| Returned Update Block Data | Description | Expected Value |
| Header Version (Offset 0) | Update Block used | 00000001h |
| Header Version (Offset 0) | Update Block unused | FFFFFFFFh |
| Offset 0Dh | Model | 1 |
| Offset 0Ch | Stepping | Valid Steppings |
| Offset 04h | Update Revision | Valid Update Revision |

*In addition, the following conditions are tested for each Update Block returned:*

i.) The checksum of the returned Update Block data is checked.

***Invalid Update Number Test***

**Description:**
This test checks to insure that the BIOS correctly returns UPDATE_NUM_INVALID when the Function 03h call is made with SI register (Update Number) = Update Cnt. The Update Cnt value represents the number of Update Blocks the BIOS claims it supports when the Function-00h Presence test is executed. Since the Update Number (number used to reference a particular Update Block in NVRAM) convention starts with 0 as the first Update Block in NVRAM, then the highest Update Number that exists for a given BIOS should be equal to Update Cnt -1.

In summary, the values in the following registers are inspected following the function 03h call. Errors are reported for any item(s) that return an incorrect response. The expected returned values are as follows:

| Register on Entry | Description | Test Parameter |
|---|---|---|
| SI | Update Number | Update Cnt from presence call |
| Register on Return | Description | Expected Value |
| CF | Carry Flag | 1 |
| AH | Return Code | 99h UPDATE_NUM_INVALID |